2014 IEEE Congress on Evolutionary Computation (CEC)
July 6-11, 2014, Beijing, China

An Evolutionary Algorithm Based on Covariance Matrix Leaning and Searching Preference for Solving CEC 2014 Benchmark Problems

Lei Chen
Guangdong University of Technology
Guangzhou, China
Zhe Zheng
Guangdong University of Technology
Guangzhou, China

Hai-Lin Liu
Guangdong University of Technology
Guangzhou, China
Email: hlliu@gdut.edu.cn
Shengli Xie
Guangdong University of Technology
Guangzhou, China
Email: shlxie@gdut.edu.cn

***Abstract*—In this paper, we propose a single objective optimization evolutionary algorithm (EA) based on Covariance Matrix Learning and Searching Preference (CMLSP) and design a switching method which is used to combine CMLSP and Covariance Matrix Adaptation Evolution Strategy (CMAES). Then we investigate the performance of the switch method on a set of 30 noiseless optimization problems designed for the special session on real-parameter optimization of CEC 2014. The basic idea of the proposed CMLSP is that it is more likely to find a better individual around a good individual. That is to say, the better an individual is, the more resources should be invested to search the region around the individual. To achieve it, we discard the traditional crossover and mutation and design a novel method based on the covariance matrix leaning to generate high quality solutions. The best individual found so far is used as the mean of a Gaussian distribution and the covariance of the best λ individuals in the population are used as the evaluation of its covariance matrix and we sample the next generation individual from the Gaussian distribution other than using crossover and mutation. In the process of generating new individuals, the best individual is changed if ever a better one is found. This search strategy emphasizes the region around the best individual so that a faster convergence can be achieved. The use of switch method is to make best use of the proposed CMLSP and existing CMAES. At last, we report the results.**

I. INTRODUCTION

Real parameter single optimisation problems have been widely studied in the past 20 years [3] [4]. In general, single objective optimization problems are often formed as a black-box problem in many fields including science, engineering, management, military, and so on. EA and other heuristic algorithms,such as PSO [8], has natural advantages in solving such kinds of problem [5]. Several global optimization algorithms have been developed and investigated. The genetic operators play an essential role in the performance of the algorithm so that the main concern is the design of genetic operators when design EA for any single objective optimisation problem. Researchers have made great efforts to combine techniques from other research fields with EAs to enhance the performance of EAs. However, if an EA treats the two parents involved in crossover equally such as randomly choose two individuals for crossover, the search would be blind. With this in mind, Gu et al. proposed a fast evolutionary algorithm with preference(FEA) [1]. They propose a scheme for choosing parents based on greedy algorithm and simulated annealing [9]. The crossover is performed between each individual and the best individual found so far to exploit the most promising regions while the simulated annealing factor is used to adjust the search range of the population. It is obvious that in the initial stage of evolution the search region should be big to escape the local optima while in the later stage the search region should be smaller in order to accelerate convergence. This strategy can make offspring generate around the best individual with high probability and it is helpful for a lot of existing test functions. They also have proofed the algorithm can converge to a global optimum in probability.

However, their method is invalid when solving the high ill-Conditioned test functions in CEC 14 test suite [2]. The crossover and mutation used in the EA are axis-parallel, which means that they don't consider the correlation of different component. Like most of the axis-parallel crossover and mutation, they are not appropriate for this kind of benchmark functions.

CMA-ES [12] which is proposed by Hansen et al. is an popular method to solve black box problems. However, when to solve multi-modality problem, CMAES may end up to premature convergence.

In this paper, we extend the FEA and combine it and CMAES to solve CEC 14 benchmark problems. We use a different method based on the covariance matrix learning to generate new individuals. It is similar to the estimation of distribution algorithms [7] [11], but also has its own character. As we all know, the covariance matrix of the best λ individuals can reflect the rotation information of the function in some extent. By learning from the population information in present generation and the accumulated information in previous generations, the evaluated covariance matrix can be utilized to guide the generation of new individuals. To make the best of the information provided by the covariance matrix, we sample new individuals from a Gaussian distribution centered the best individual found so far in each generation. The best individual of the population is used to lead the population evolution. Once a better individual than the present best individual is generated, it immediately used as the new center for sampling new individual. In the process of evolution, the constantly updated covariance matrix and center individual guide the direction of the evolution. In other words, individuals are always generated centered the best individual which is effective to exploit the most promising area in search space. The improved strategy guarantees the high quality of the new individuals.

The remainder of this paper is organized as follows: Section II presents the main idea of the proposed EA based on covariance matrix leaning and searching preference in details. Section III gives the framework of proposed switch method. In section IV, we study the performance of the proposed method using the CEC 14 test suite and report the results. Section V concludes the paper.

II. The Main Idea of Proposed Algorithm Based on Covariance Matrix Leaning and Searching Preference

In the proposed EA, the best individual found so far in the process of evolution is used as the leader to guide the search. The newly generated individuals should center the best individual. It is reasonable to think that the region around the best individual is the potential region to find the next better individuals. Therefore the searching preference is realized by using this method. For single objective optimization problem which is highly ill-condition, the traditional axis-parallel crossover and mutation are not suitable. Because of the ill-condition rotation matrix, the traditional crossover and mutation cannot generate individual with high quality [12]. If we know the rotation matrix, we can find the optima easily [10]. However, CEC 14 test benchmark functions are all black-box function and it is impossible to identify the rotation matrix. We can only use the population distributes information to evaluate the rotation matrix.

A. The Covariance Matrix Leaning strategy

Although the idea of evaluating the covariance matrix is obvious, the accuracy of the evaluation can be very difficult to guarantee. The accuracy can directly affect the performance of the algorithm when to solve this kind of ill-condition test functions. Considering the importance of the accuracy, we design the covariance matrix leaning strategy. The strategy can make the covariance matrix evaluation more accurate by learning from both previous population and the present population. By using this strategy, the covariance matrix inherits the information accumulated in the process of evolution and also learns new information from the present population. The covariance matrix is calculated by using the following equation.

$$C = \sum_{i=1}^{t*\lambda} (x_i - x^{best})' * (x_i - x^{best}) \tag{1}$$

where, $t * \lambda$ means the best λ individuals in precious t generations which are used to evaluate rotation matrix. x_{center} is the present best individual which is updated if a better individual is generated during the evolution.

B. The Generation of New Individuals

A total of $popsize$ new individuals are generated in every generation, where the $popsize$ represents the population size. As we have stressed, the traditional axis-parallel crossover and mutation are not suitable for the ill-condition test functions. So the estimation of distribution strategy is used to replace the mutation and crossover to generate the new individuals. The next generation population is generated from a gaussian distribution as follows:

$$x_i^{g+1} = x^{best} + \sigma * N(0, C^g), i = 1, 2, ..., popsize \tag{2}$$

where $\sigma = r_1(1 - rand^{(1-\frac{fitcount}{MaxFES})^{0.7}})$ is the search step size which is similar to simulated annealing algorithm [9] gradually decreasing the search range. From equation 2, we know that σ tends to 0 as g tends to Max_{gen}. r_1 is the random number in $[0, 1]$, g is the current generation number and Max_{gen} is the maximum generation number. x^{best} is the best individual in the population, which is used to lead the evolution of the population. In order to make the search more effective, the x^{best} is changed once a better individual is found.

If kth component v_{ik} of v_i is out of the boundary, namely, if v_{ik} smaller than the low boundary l_k^D,

$$v_{ik} = l_k^D + 0.5rand(l_k^D - v_{ik}).$$

If v_{ik} is bigger than the upper boundary, for example, the kth component v_{ik} of v_i is out of l_k^U,

$$v_{ik} = l_k^U - 0.5rand(l_k^U - v_{ik}).$$

where $rand$ is a random number in [0,1].

C. The Update of The Population

When $popsize$ new individuals are generated, the population need to be updated. In order to prevent premature and maintain population diversity, an new update strategy is designed. Firstly, new-generated individuals and the present population are mixed together and made pairs randomly; then the better one of every pair of individuals is selected as the next generation individual.

III. The Main Idea Of The Switch Method

In this section, we will give the basic idea of the switch method. For the sake of readability, the proposed CMLSP are given in detail firstly.

The main framework of proposed algorithm CMLSP is as follows(**Algorithm1**):

The switch method works as follows(**Algorithm2**):

IV. Experimental Studies

The simulation program has been developed within the Matlab7.5 programming environment.

Algorithm 1: CMLSP

Input :
- single objective problem: SOP;
- maximum number of function evaluations: $MaxFES$;
- population size: $popsize$
- $\lambda = popsize/2$

Output: The best individual in the final population

Initialization: Uniformly randomly initialize the population Q within the search space and generate $popsize$ individuals, calculate their $F - value$(i.e objective value) in objective space and find the best individual x^{best} in the population. Set the current generation: $gen = 1$, find the best λ individuals in the population and calculate its covariance matrix C, $fitcount = popsize$, $AS = \varnothing$.
while $fitcount < MaxFES$ **do**
  Generation of New Solutions: Set $R = \emptyset$;
  for $i \leftarrow 1$ **to** $popsize$ **do**
    generate new individual $x_i = x^{best} + \sigma * N(0, C)$
    Compute $F(x_i)$
    if $F(x_i) < F(x^{best})$ **then**
      $x^{best} = x_i$;
      $R := R \cup \{x^{best}\}$;
      update C by Eq.1;
    else
      $R := R \cup \{x^i\}$;
    end
  end
  $fitcount = fitcount + popsize$;
  update the population Q using R;
  select λ best individuals in Q;
  use Q to update C by Eq.1
end

A. Experimental Setting

- Problem: 30 minimization problem for CEC 14 Special Session.
- Dimensions: $D = 10, 30$.
- Runs / problem: 51
- $popsize$: 100
- MaxFES: 10000*D
- Search Range: $[-100, 100]^D$
- $N1 = 10 * D, N2 = 5 * D$

B. Computational Complexity

The results of experimental runs on Function 18 are given in Table III according to [2].

C. Results

The results for each function and problem dimension are given according to [2] in Tables I-II after the maximum number of function evaluations.

Algorithm 2: switch method

Input :
- single objective problem: SOP;
- the number of function evaluations to decide if CMAES is changed: $N1$;
- the number of function evaluations to decide if CMLSP is changed: $N2$

Output: The best individual in the final population

Initialization: $index = 1$;
while $fitcount < MaxFES$ **do**
  Switch $index$;
  Case 1;
  Run CMAES;
  if *the best individual doesn't change after* $N1$ *function evaluations* **then**
    $index = index * (-1)$
  end
  Case -1;
  Run CMLSP;
  if *the best individual doesn't change after* $N2$ *function evaluations* **then**
    $index = index * (-1)$
  end
end

TABLE I. COMPUTATIONAL COMPLEXITY

	$T0$	$T1$	$\widetilde{T2}$	$(\widetilde{T2} - T1)/T0$
D=10	0.192657	1.358225	2.4078	5.4480
D=30	0.192657	2.013336	4.2803	11.7669
D=50	0.192657	3.208305	6.9630	19.4888

V. CONCLUSION

In this paper, we propose an EA based on covariance matrix leaning and searching preference and combine it with CMAES for CEC 2014 test suite. The basic idea of the proposed CMLSP is simple but effective: the better an individual is, the more resources are invested to search the regions around to the individual. A new strategy based on covariance matrix learning is designed to generate new individuals. A switch method is designed to make best use of CMLSP and CMAES. We have reported the results by using the switch method to solve 30 noiseless optimization problems designed for the special session on real-parameter optimization of CEC 2014. The results demonstrate the efficiency of the proposed CMLSP.

ACKNOWLEDGMENT

This work was supported in part by the Natural Science Foundation of Guangdong Province (S2011030002886, S2012010008813), and in part by the projects of Science and Technology of Guangdong Province (2012B091100033).

REFERENCES

[1] Gu, F., Liu, H., Li X, "A Fast Evolutionary Algorithm with Searching Preference", Int. J. Computation Science and Engineering, Vol. 1, Nos. 3/4, (2012): 197-212.

[2] Liang J J, Qu B Y, Suganthan P N, " Problem Definitions and Evaluation Criteria for the CEC 2014 Special Session and Competition on Single Objective Real-Parameter Numerical Optimization, 2013.

TABLE II. RESULTS FOR 10D

Func.	Best	Worst	Median	Mean	Std
1	0	7.16653E-06	0	3.48643E-07	1.32420E-06
2	0	0	0	0	0
3	0	4.72417E-04	2.24436E-05	5.56217E-05	8.41130E-05
4	0	0	0	0	0
5	0	1.99999E+01	1.99999E+01	1.56863E+01	8.22577E+00
6	0	2	0	2.94898E-02	2.07962E-01
7	0	0	0	0	0
8	0	2.65932E+01	9.94959E-01	2.10690E+00	3.64861E+00
9	0	6	9.94959E-01	1.53541E+00	1.24459E+00
10	2.49818E-01	3.62207E+02	2.43707E+02	2.11360E+02	7.36555E+01
11	3.60232E+00	3.70435E+02	1.33620E+02	1.43167E+02	1.09246E+02
12	0	2.41366E-01	0	4.64975E-02	7.97516E-02
13	7.37144E-03	4.96121E-02	2.32101E-02	2.46556E-02	9.49529E-03
14	4.85026E-02	3.70116E-01	1.94779E-01	1.98300E-01	5.90713E-02
15	4.50828E-01	1.44881E+00	9.10095E-01	8.93853E-01	2.40492E-01
16	2.62208E-01	2.66797E+00	1.63403E+00	1.59807E+00	6.44282E-01
17	2.19806E+00	6.96702E+02	3.04921E+02	3.03885E+02	1.90532E+02
18	2.20576E-01	6.51046E+01	3.25093E+01	3.23827E+01	1.33587E+01
19	2.91916E-02	1.54751E+00	1.49493E+00	1.28326E+00	4.33040E-01
20	2.11254E-01	7.18289E+01	2.17121E+01	2.30820E+01	1.49655E+01
21	1.16272E-02	3.63019E+02	1.77776E+01	5.00116E+01	7.17984E+01
22	4.42334E-01	1.62879E+02	2.73167E+01	5.89947E+01	5.20786E+01
23	2.00000E+02	2.00000E+02	2.00000E+02	2.00000E+02	0
24	1.00000E+02	1.49327E+02	1.08494E+02	1.11975E+02	1.11933E+01
25	1.00000E+02	1.68866E+02	1.18539E+02	1.24516E+02	1.84115E+01
26	1.00005E+02	1.00049E+02	1.00018E+02	1.00018E+02	8.32068E-03
27	3.59540E-01	5.81730E+01	1.58245E+00	2.80464E+00	7.85062E+00
28	1.00020E+02	2.00000E+02	2.00000E+02	1.79753E+02	3.96047E+01
29	2.00000E+02	2.00000E+02	2.00000E+02	2.00000E+02	0
30	2.00000E+02	2.00000E+02	2.00000E+02	2.00000E+02	4.51228E-05

TABLE III. RESULTS FOR 30D

Func.	Best	Worst	Median	Mean	Std
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	0	0
4	0	0	0	0	0
5	1.99989E+01	1.99990E+01	1.99990E+01	1.99990E+01	2.28071E-05
6	0	0	0	0	0
7	0	0	0	0	0
8	2.98488E+00	1.08450E+02	7.95967E+00	1.03313E+01	1.47652E+01
9	2.98488E+00	2.98488E+01	7.95967E+00	8.99640E+00	4.23486E+00
10	1.23121E+02	3.42016E+03	1.35624E+03	1.55996E+03	8.97182E+02
11	6.17392E+02	4.01893E+03	2.61797E+03	2.41276E+03	9.08234E+02
12	0	2.12112E-02	2.93010E-03	4.33060E-03	5.37717E-03
13	2.73467E-02	1.15493E-01	5.59640E-02	5.86279E-02	2.01092E-02
14	2.05081E-01	3.75061E-01	3.15259E-01	3.11185E-01	3.79531E-02
15	1.18936E+00	3.91331E+00	3.11361E+00	3.07240E+00	4.81567E-01
16	7.73028E+00	1.26026E+01	1.14908E+01	1.13712E+01	7.96936E-01
17	1.41363E+01	1.80414E+03	1.07525E+02	2.59911E+02	4.07754E+02
18	4.31010E-01	2.54850E+02	9.27363E+01	9.48095E+01	5.25574E+01
19	2.62822E+00	9.08196E+00	5.46993E+00	5.68318E+00	1.78472E+00
20	2.12894E+00	1.97228E+02	1.83160E+01	3.86554E+01	4.92278E+01
21	2.18693E+00	1.12824E+03	5.98898E+02	5.72834E+02	2.85305E+02
22	2.08573E+01	4.13284E+02	1.40632E+02	1.18195E+02	9.49732E+01
23	2.00000E+02	2.00000E+02	2.00000E+02	2.00000E+02	0
24	2.00000E+02	2.00001E+02	2.00000E+02	2.00000E+02	8.74542E-05
25	2.00000E+02	2.00000E+02	2.00000E+02	2.00000E+02	0
26	1.00098E+02	1.00438E+02	1.00173E+02	1.00181E+02	5.12956E-02
27	2.00000E+02	2.00000E+02	2.00000E+02	2.00000E+02	0
28	2.00000E+02	2.00000E+02	2.00000E+02	2.00000E+02	0
29	2.00000E+02	2.00000E+02	2.00000E+02	2.00000E+02	0
30	2.00000E+02	2.00000E+02	2.00000E+02	2.00000E+02	0

[3] Back, Thomas, and Hans-Paul Schwefel, "An overview of evolutionary algorithms for parameter optimization." Evolutionary computation 1.1 (1993): 1-23.

[4] Grefenstette, John J, "Optimization of control parameters for genetic algorithms." Systems, Man and Cybernetics, IEEE Transactions on 16.1 (1986): 122-128.

[5] Whitley, Darrell, "An overview of evolutionary algorithms: practical issues and common pitfalls." Information and software technology 43.14 (2001): 817-831.

[6] Edmonds, Jack, "Matroids and the greedy algorithm." Mathematical programming 1.1 (1971): 127-136.

[7] Lee, Chang-Yong, and Xin Yao, "Evolutionary programming using mutations based on the Lvy probability distribution." Evolutionary Computation, IEEE Transactions on 8.1 (2004): 1-13.

[8] Mo, Simin, Jianchao Zeng, and Ying Tan, "Particle swarm optimisation based on self-organisation topology driven by different fitness rank." International Journal of Computational Science and Engineering 6.1 (2011): 24-33.

[9] Rajan, C. Christober Asir, "Genetic Algorithm Based Simulated Annealing Method for Solving Unit Commitment Problem in Utility System." AIP Conference Proceedings. Vol. 1298. 2010.

[10] Andrews, George Eyre. q-Series: Their development and application in analysis, number theory, combinatorics, physics and computer algebra. No. 66. AMS Bookstore, 1986.

[11] Larranaga, Pedro, and Jose A. Lozano, eds. Estimation of distribution algorithms: A new tool for evolutionary computation. Vol. 2. Springer, 2002.

[12] Hansen, Nikolaus, et al, "Real-parameter black-box optimization benchmarking 2010: Experimental setup." (2010).